Implementing IPSec for Embedded Devices

Robert Vamosi, Senior Analyst, Mocana

With millions of new electronic devices connecting to the Internet every day, criminal hackers are increasingly focusing on a new type of target: mobile and embedded systems. Such systems include point-of-sale terminals, wireless routers, smart phones, our cars, and the emerging medical technical infrastructure. Perhaps the most ominous of the new hacking trends is the upsurge in cyberattacks against our utility infrastructure. But there’s a robust security approach that can mitigate a lot of these problems. It’s called IPSec. But it’s not used much on embedded devices.

Internet Protocol Security or "IPSec" is used to secure the confidentiality, integrity, and authentication of the data sent between two machines. It can restrict which computers communicate with each other, provide message integrity by encrypting the shell and not necessarily the data inside, and provide mutual authentication by permitting requests only from a specific client or Web server. IPSec is also terrific for reducing the threat of packet sniffers, man-in-the-middle attacks or replay attacks. Additionally, IPSec can defend against network attacks by using packet filtering and trusted communication enforcement. Most IPSec is implemented on PCs, workstations and network appliances. Even today, in 2011, there aren’t a lot of embedded devices that can "speak IPSec".

Why? Well, for one thing, IPSec does not dictate specific authentication and encryption methods, and that can make implementing it more complex. Additionally, there just aren’t a lot of IPSec packages that will work on embedded systems. Most IPSec implementations are too big, too slow, or too power-hungry to be appropriate for devices and the "Internet of Things." That’s unfortunate, because currently there are at least five times as many connected devices on the Internet as PCs, and almost none of these devices are capable of protecting their own traffic.

As attackers increasingly target embedded systems, there is an urgent need to secure all these devices, not just PCs or network appliances. Embedded devices need IPSec, too.

A Quick IPSec Primer

IPSec is a suite of protocols operating at Layer 3, the network layer. SSL and SSH, which also provide authentication and encryption of the communications between two systems, operate one layer higher. By being further down the stack, IPSec is transparent to applications. Applications continue to communicate with one another in the normal manner using TCP and UDP ports. Developed alongside IPv6, and mandatory for fully compliant IPv6 implementations, IPSec is still most often implemented as an add-on to IPv4.

IPSec is a framework, i.e. it does not dictate the use of specific hashing and encryption algorithms. Therefore the IPSec implementation that want to communicate with someone new has to (1) have several different security methods

“at the ready” and (2) have a way of communicating its capabilities to its potential comms partner before the session is set up. That’s why each device has at least one Security Association (SA), which stores the list of parameters the device can use when communicating, such as the authentication and encryption keys, algorithms, key lifetime, and source IP addresses.

For its authenticating protocol, IPSec uses the Authentication Header (AH) technique, and for its authenticating and encrypting protocol, Encapsulating Security Payload (ESP). If a company only needs to ensure the integrity of the content it would use AH. If a company also needs to have confidentiality it would use ESP instead.

Tunnel and Transport Mode

One of the most common uses for IPSec is for virtual private networks. IPSec's tunnel mode, where the entire packet is encrypted, is used to create secure virtual private networks for network-to-network communications, host-to-network communications, and host-to-host communications such as private chat. Once a VPN is established, the two ends can run virtually any data, voice or video application across it, securely.

In Transport mode only the payload is encrypted and/or authenticated. Transport mode within IPSec provides host-to-host communications, although it can be used for network-to-network and host-to-network as well.

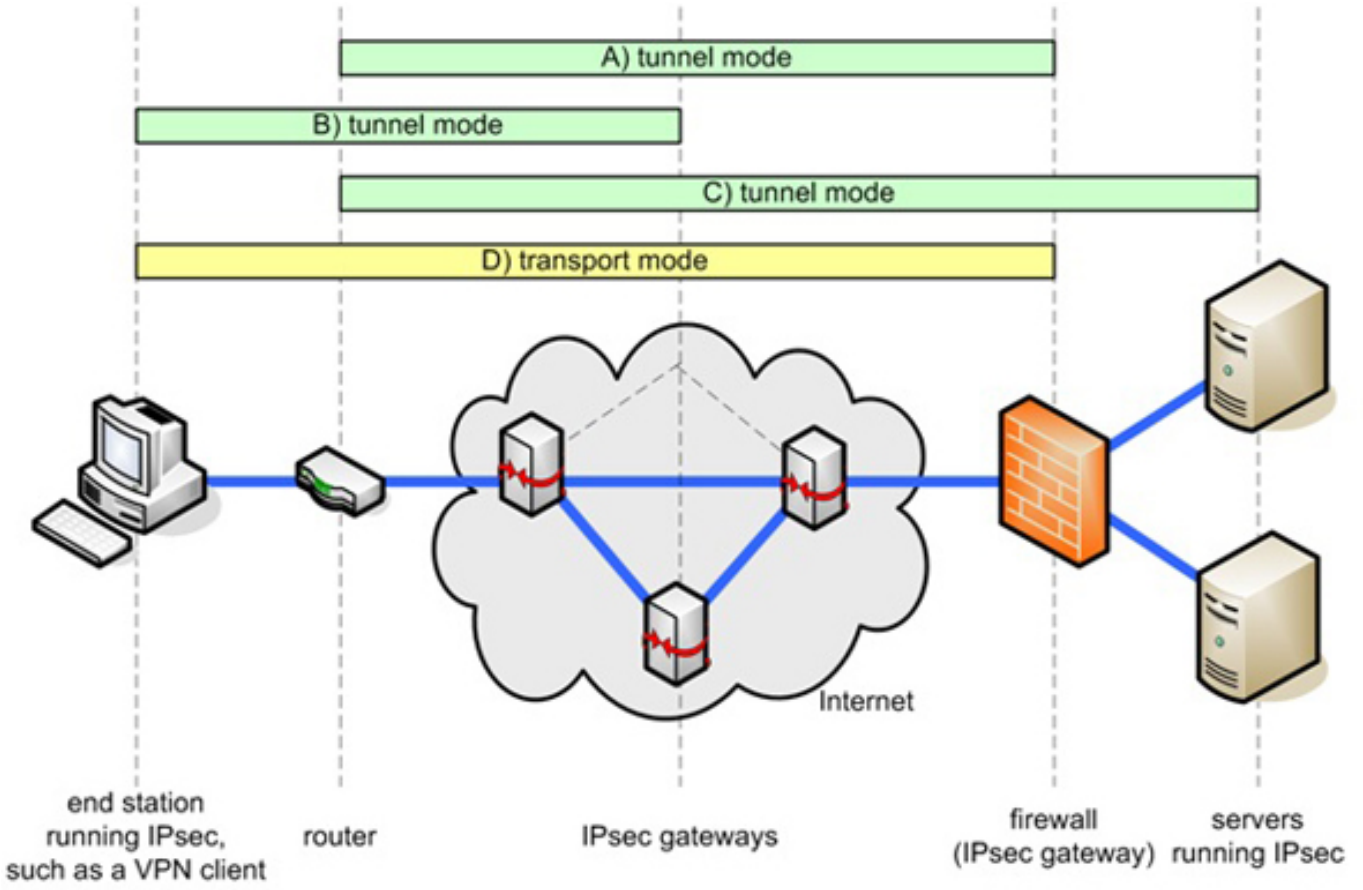

Figure 1. IPSec tunnel and transport modes

Mobile environments
The fastest growing security market is the mobile device ecosystem. IPSec's Mobile IKE (MOBIKE) is specifically designed for scenarios where one of the two end points keeps changing IP addresses. A mobile device Virtual Private Network (VPN) client could use MOBIKE to keep a secure connection with the VPN gateway active, even while repeatedly moving from one IP address to another.

A user could start from fixed Ethernet at her desk, then disconnect the laptop from its docking station and move to the office's conference room wireless LAN—all while remaining securely connected. At the end of the day when the user leaves the office, the laptop could start using a 3G or 4G cellular connection for the train ride home. Then, when the user finally arrives home, the laptop could switch to the home wireless LAN. A well-implemented MOBIKE app can guarantee that, as long as connectivity is continuously available, a secure VOIP call or videoconference maintained without interruption, across all the "physical" network transitions.

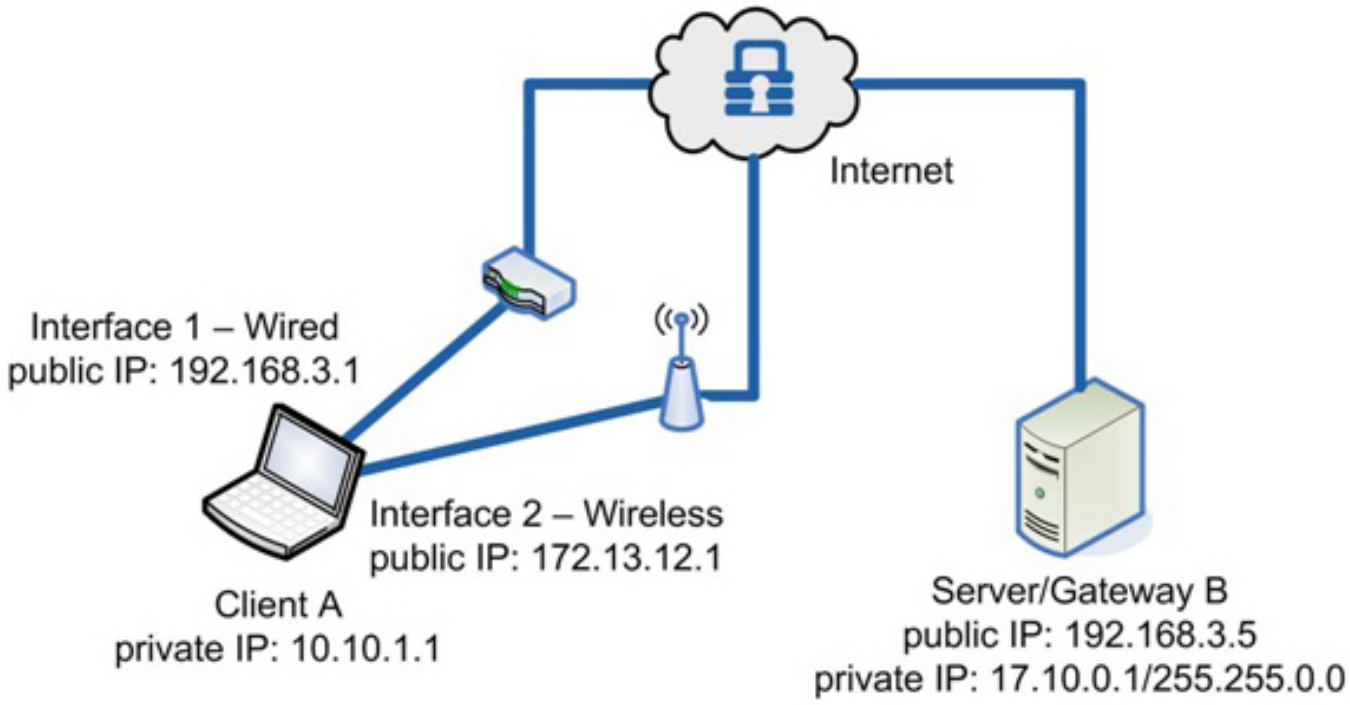

Figure 2. Example of using MOBIKE to switch between a laptop's two interfaces

Applying IPSec Across Device Types
But by far, most of the new demand for IPSec will be for devices that aren't PCs or Laptops. Remote access VPN connections are already de rigeur for enterprises and the government, as is robust encryption and authentication inside these connections. So its natural to project that users (especially enterprise and government) will demand IPSec functionality for their smartphones, tablets and consumer electronics devices as well.

Wireless Internet access is quickly becoming standard on airlines worldwide. New cars, trucks and bus models contain wireless connectivity features for their occupants. When compromised, customers will suffer fraud, abuse or robbery of

their personal information or credit card numbers.

In consumer electronics, manufacturers of consumer electronics are rushing to make their portfolios of products "internet ready", but need to make sure that their TV, toy or gadget doesn't become the "weakest link" in the home network – and the vector for attacks against their customers.

In Industrial Automation, remote access to factory floor and SCADA equipment can quickly turn dangerous to life and property if that access is compromised by malware or hackers. In smartmeters and ICS systems, dozens of communications protocols are part of the worldwide rollout of the Smart Grid; here, secure remote access capability is a given.

On a more personal level, more than 25 million Americans have some form of implanted medical device. Increasingly, those devices communicate remotely with servers and controllers. Here, security can be a matter of life or death. So you see, implementing IPSec down at the embedded device level isn't just a "nice to have". For many device applications – particularly those that operate in critical contexts - IPSec will soon be expected as a standard, out-of-the-box capability.

Using IPSec in Embedded Systems
As mentioned, there are a number of challenges to implementing IPSec on resource-constrained embedded systems. Many open source implementations are either too large, or too resource intensive for smart meters in the home. Therefore, look for a proprietary solution with the following:

• Compact size: An IPsec solution should fit into tiny memory footprints. Often open-source packages can't match the performance optimization found in paid packages. Solutions that offload IPsec and IKE crypto operations from the main CPU can deliver significant performance enhancements.

• FIPS Certified with NSA Suite B Support: All government agencies and most contractors now require FIPS-certification of cryptographic engine —a difficult certification to achieve. And the binary should have full support for NSA's Suite B algorithms, providing secure communications between high-assurance (classified) and basic-assurance systems.

• A Complete Solution: An embedded IPsec solution should be not missing critical standards, algorithms or code needed to finish your IPsec/IKE implementation.

• No General Public License code: This gives confidence that your intellectual property won't accidentally become public domain because of "GPL contamination"—something open source projects can't do.

• Support a Variety of Platforms: An embedded IPsec solution should at least work with Linux, Windows, VxWorks, ThreadX and QNX.

Robert Vamosi is a senior analyst Mocana and the author of When Gadgets

Betray Us: The Dark Side of Our Infatuation with New Technologies.